Distributed Multimedia and QOS: A Survey

Andreas Vogel
CRC for Distributed Systems Technology

Brigitte Kerhervé
Université de Quebéc á Montréal

Gregor von Bochmann and Jan Gecsei
Université de Montréal

Quality of service is increasingly important for all components within distributed multimedia systems, as this survey reveals. We discuss QOS parameters found in communication protocols,operating systems, multimedia databases, and file servers, as well as those directly affecting the human user.

The notion of quality of service originally emerged in communications to describe certain technical characteristics of data transmission. For example, the Open Systems Interconnection (OSI) Reference Model has a number of QOS parameters describing the speed and reliability of transmission, such as throughput, transit delay, error rate, and connection establishment failure probability. These parameters apply mostly to lower protocol layers and are not meant to be directly observable or verifiable by the application. Consequently, OSI's QOS coverage is incomplete and even inconsistent. This situation, while acceptable when communication networks were used mostly for non-time-dependent data, is no longer satisfactory with the new requirements stemming from distributed multimedia systems. As time-dependent data become prevalent in multimedia applications, the entire distributed system must participate in providing the guaranteed performance levels. In this view, an application process originates the QOS requirements and conveys them in the form of QOS parameters to other system components. Generally, a negotiation process among the components of the system then determines if collectively they can satisfy the requested QOS level.

What is QOS?

Beyond its intuitive meaning as system characteristics that influence the perceived quality of an application, there is little consensus on the precise meaning, let alone the formal definition, of QOS. For example, the Reference Model for Open Distributed Processing, or RM-ODP, refers to QOS as "A set of quality requirements on the collective behavior of one or more objects."[1] This definition is too general to be meaningful, since it includes all system parameters without distinction.

For this survey, we use the following working definition:

> Quality of service represents the set of those quantitative and qualitative characteristics of a distributed multimedia system necessary to achieve the required functionality of an application.

Functionality includes both the presentation of multimedia data to the user and general user satisfaction. The QOS of a given system is expressed as a set of (parameter-value) pairs, sometimes called a tuple; we consider each parameter as a typed variable whose values can range over a given set.

Different applications on the same distributed system can have different subsets of relevant QOS parameters, with different values required, and some parameters might not be mutually independent. In this survey, we use the term parameter in two senses: as the parameter itself (such as throughput) or as a parameter-value pair (such as packet loss rate = 10^{-9}). In a distributed multimedia system, it is hard to separate the QOS parameters from other system parameters. However, one distinguishing feature is that QOS parameters are subject to negotiation between system components.

Distributed multimedia applications can be presentational or conversational, although most applications have both presentational and conversational aspects. Presentational applications provide remote access to multimedia documents such as video-on-demand services, while conversational applications such as computer-supported cooperative work (CSCW) typically involve real-time multimedia communication. Conversational applications can be further classified into on-demand and broadcast services. The application type has a decisive influence on the required system parameters. For example, delay (see Table 1) is less important for presentational applications than for conversational ones.

1070-986X/95/$4.00 © 1995 IEEE

Simplified QOS and complicating factors

Processing QOS in a distributed multimedia system involves several related activities:

1. Assessing the QOS requirements in terms of users' subjective wishes or satisfaction with the quality of the application—performance, synchronization, cost, and so forth.

2. Mapping the assessment results onto QOS parameters for various system components or layers. For example, the user chooses video in terms of its resolution and frame rate, which map onto throughput requirements.

3. Negotiating between system components or layers (embedded in protocols) to ensure that all system components can meet the required parameters consistently.

If the negotiation ends with an agreement on the required values, the application can be launched. Types of agreements include guaranteed, best-effort, or stochastic.

We can complicate this simplified QOS processing model by considering some additional issues. For example, QOS requirements may change during an application session. A medical teleconsultation using low-quality video might entail showing a series of high-quality X-ray images at one point; this requires QOS renegotiation to increase the bandwidth for the X-ray images. Also, sometimes the negotiated parameters cannot be maintained due to network congestion, requiring renegotiation.

Verifiable mappings between architectural layers are generally not one-to-one. Some parameters are mutually dependent or contradictory; for example, decreasing the error rate by permitting retransmission increases the average transit delay. Further, in practice, the required QOS values correspond not to a well-defined point, but to a region in the parameter space; the instantaneous working point within this region can change over time.

In spite of the contract resulting from QOS negotiation, the actual QOS values in the system can also vary over time. Changing system load can trigger adjustments in the transport subsystem or in the operating system. Therefore, the system must continuously monitor the actual QOS and employ correction mechanisms such as blocking lower priority tasks. In this perspective, maintaining QOS becomes a complex control problem.

Table 1. The five categories of QOS parameters.

Category	Example Parameters
Performance-oriented	End-to-end delay and bit rate
Format-oriented	Video resolution, frame rate, storage format, and compression scheme
Synchronization-oriented	Skew between the beginning of audio and video sequences
Cost-oriented	Connection and data transmission charges and copyright fees
User-oriented	Subjective image and sound quality

The user interface

In our view, people are the starting point for overall QOS considerations. Thus the primary source of QOS requirements is the user, and a suitable interface should be provided to facilitate the choice of parameters. Until recently, this view has not been sufficiently emphasized in the literature (see the "User issues" sidebar).

A general discussion of the user's perspective introduced the "Quality Query by Example."[2] The essence of this method is to hide, as much as possible, the internal system QOS parameters (often meaningless to the user) and to present instead a choice from examples of varying quality, such as images of different size, resolution, and color depth, or speech of telephone or CD quality. The user choices are automatically mapped into system parameters. The interface also should memorize user profiles to avoid making the user repeat the lengthy selection process. While this method is suitable for presentational parameters such as

User Issues

Steinmetz and Engler discussed user involvement in synchronization issues.

R. Steinmetz and C. Engler, "Human Perception of Media Synchronization," Tech. Report 43.9310, IBM European Networking Center, Heidelberg, 1993.

Apteker et al. investigated the relationship between user acceptance and QOS degradation.

R.T. Apteker et al., "Distributed Multimedia: User Perception and Dynamic QOS," *Proc. IS&T/SPIE Symp. on Electronic Imaging: Science & Technology, Workshop on High-Speed Networking and Multimedia Computing*, SPIE, New York, 1994, pp. 226-234.

Figure 1. In this quality query by example, the user selected a high-quality, full-size, color version of the image.

video, images, and sound, it is less adequate for parameters such as response time or synchronization, which could require a more direct specification better served by a slider.

In a realistic system the choices are not independent. Selecting a high-resolution image might incur increased cost and delivery delay. Users should know the ramifications before they make their selections, to prevent them from automatically selecting the best available qualities without regard to consequences.

The real QOS choices available to the user depend on all system components: the operating system (lack of real-time capability might limit the precision of synchronization), the transport system (a slow segment might limit the throughput), or the application (the database might contain only low-quality images). Figures 1 and 2 demonstrate QOS parameters of different image qualities using a variation of the quality query by example method. While Figure 1 shows a full-size, high-resolution, color version of an image, Figure 2 demonstrates lower quality—an iconized, low-resolution, black-and-white version.

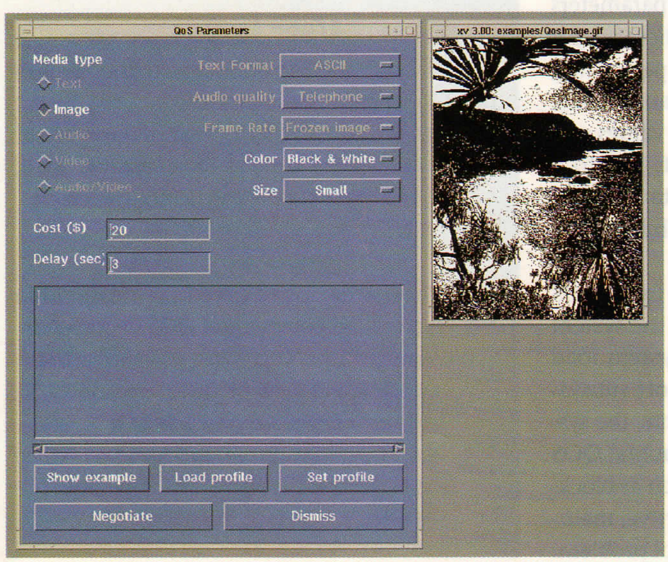


Figure 2. In this quality query by example, the image's QOS parameters are more modest—black and white, low resolution, and small size.

End systems

The parameters of end systems can have a strong impact on the QOS the user perceives. For example, the CPU and bus speed can limit the frame rate of the video presentation, and a black-and-white screen cannot display color images. However, such "hidden" parameters are taken into account through the QOS parameters of the operating system.

The basic QOS constraints on operating systems relate to their real-time behavior. Herrtwich[3] gave an overview of the real-time requirements placed on operating systems to satisfy multimedia applications; critical issues include performance, scheduling, and resource reservation.

Standard Unix systems generally do not meet these requirements. For example, one report concluded, "For real-time and multimedia systems that are limited by the worst-case performance, Mach imposes a very high overhead."[4] Solutions to these problems include extending existing operating systems or reimplementing Unix systems.

The most common approach is extensions. Real-Time (RT) Mach "extends Mach 3.0 with real-time threads and scheduling, which should greatly enhance our low-latency applications."[5]

Other adaptations take advantage of microkernel architectures. Mercer et al.[6] developed a processor reservation strategy specifically designed for the microkernel architecture and implemented for RT Mach. Nakajima et al.[4] presented a similar approach with a real-time RT Mach server for predictable services.

Extensions do not always solve all the problems, however; as an example, Nahrstedt and Smith[7] found the real-time extended AIX only partially suitable for multimedia applications.

An example for operating systems outside the Unix world is OS/2.

Parsons[8] reported a multimedia architecture based on OS/2 whose real-time capabilities satisfy multimedia applications.

We can identify operating-system-related QOS parameters at different abstraction levels. Low-level parameters include performance, scheduling, and size of available main and virtual memory. On a more abstract level, the operating systems provide the QOS of certain services, such as the throughput and delay of an MPEG player. Such a high-level view provides a better base for an overall QOS negotiation.

The impact of encoding

The data encoding method influences QOS parameters, particularly for video. We can classify video coding schemes into a hierarchy:

1. Intraframe compression
2. Intra- and interframe compression
3. Layered compression

The first level contains coding schemes that use intraframe coding, in which each frame is compressed and coded independently, such as Moving JPEG.[9] Such coding methods allow QOS variations only by decreasing the frame rate through frame dropping. Various dithering algorithms can also decrease the original encoded quality.

The second level contains schemes that use both intraframe and interframe coding, like MPEG and the ITU H.261 standard for video telephony.[10] This level of coding allows more sophisticated approaches, in particular interaction with a transport system. For example, Delgrossi et al.[11] suggested sending the I, P, and B frames of an MPEG-coded video over streams with different priorities. I frames, which contain intraframe coding, have the highest priority. The high-priority stream might receive guaranteed QOS service, while the lower priority streams get best-effort QOS.

The third level contains so-called layered or scalable coding schemes, such as those in the sidebar "Coding schemes." The idea here is to encode video in different layers, where the lowest layer contains basic frame information such as luminance, while higher layers carry additional information such as chrominance or extra bits for increased resolution. Third-level schemes allow optimization of the quantity of data transmitted.

Coding Schemes

Tawbi et al. discussed some aspects of the relationship between video compression standards and QOS, while Le Gall presented an overview of coding and compression standards.

W. Tawbi et al., "Video Compression Standards and Quality of Service," *The Computer J.* (special issue on multimedia), Vol. 36, No. 1, Feb. 1993, pp. 41-54.

D. Le Gall, "A Video Compression Standard for Multimedia Applications," *Comm. ACM*, Vol. 34, No. 4, Apr. 1991, pp. 46-58.

Girod reviewed coding schemes, promoting resolution hierarchies as a way to build a scalable video code.

B. Girod, "Scalable Video for Multimedia Workstations," *Computer and Graphics*, Vol. 17, No. 1, 1993, pp. 269-276.

The Gaussian and Laplacian Pyramid uses layering to control the size of the video.

T. Chiueh and R.H. Katz, "Multi-Resolution Video Representation for Parallel Disk Arrays," *Proc. ACM Multimedia 93*, ACM Press, New York, 1993, pp. 401-409.

Thus if the receiving workstation has a black-and-white screen, only a basic layer needs to be transmitted and decoded.

Communication protocols

The protocol hierarchy offers three levels of QOS:

1. Lower layer protocols
2. Network and transport protocols
3. Application-layer protocols

Low-level protocols such as Asynchronous Transfer Mode (ATM) and Fiber Distributed Data Interface (FDDI), managed by low-level QOS parameters, provide sufficient bandwidth and acceptable delay for multimedia traffic. Network, transport, and session layer protocols provide mechanisms for handling QOS over heterogeneous networks, mapping QOS parameters from higher to lower layers. High-layer protocols support an overall QOS negotiation between all involved components of a distributed multimedia application.

Lower layer protocols

Because of its inherent nondeterminism and rapid degradation at high utilization rates, Ethernet does not allow resource reservation. Token ring technologies such as FDDI can, accord-

File Servers

Besides the database, another important element of a distributed multimedia system is the continuous-media file server. An essential part of presentational applications, it does not necessarily intervene in conversational applications. Problems at this level are similar to those in operating systems, namely real-time scheduling, guaranteed throughput, and delays (see "End systems," main text). A central issue is organizing the disk layout to allow efficient access to continuous data:

F.A. Tobagi et al., "Streaming RAID: A Disk Array Management System for Video Files," *Proc. 1st ACM Int'l Conf. on Multimedia*, ACM Press, New York, 1993, pp. 393-400.

D. Kandlur, M.S. Chen, and Z.Y. Shae, "Design of a Multimedia Storage Server," *Proc. IS&T/SPIE Symp. on Electronic Imaging: Science & Technology, Workshop on High-Speed Networking and Multimedia Computing*, SPIE, Bellingham, Wash., pp. 164-178.

J.K. Dey, C.S. Shih, and M. Kumar, "Storage Subsystem in a Large Multimedia Server for High-Speed Network Environments," *Proc. IS&T/SPIE Symp. on Electronic Imaging: Science & Technology, Workshop on High-Speed Networking and Multimedia Computing*, SPIE, Bellingham, Wash., pp. 200-211.

The University of California at Berkeley, the University of California at San Diego, and the University of Lancaster produced notable continuous-media file server projects.

D.P. Anderson, Y. Osawa, and R. Govindan, "A File System for Continous Media," *ACM Trans. on Information Systems*, Vol. 10, No. 1, Jan. 1992, pp. 51-90.

P.V. Rangan and H.M. Vin, "Efficient Storage Techniques for Digital Continuous Multimedia," *IEEE Trans. on Knowledge and Data Eng.*, Vol. 5, No. 4, Aug. 1993, pp. 564-573.

P. Lougher and D. Sheperd, "The Design of a Storage Server for Continous Media," *The Computer J.*, Vol. 36, No. 1, Feb. 1993, pp. 32-42.

ing to their token control policy, bound the maximum delay and reserve resources for guaranteed throughput.

ATM, perhaps the lower level protocol best suited to distributed multimedia applications, provides explicit facilities for handling QOS within the signaling protocol.[12,13] To this end, the Setup and Connect messages include the information elements End-to-End Transit Delay and ATM User Cell Rate.

Network and transport protocols

Data transport requirements in the premultimedia era mainly aimed for fair and uncorrupted delivery, largely satisfied by TCP/IP and the ISO transport protocols. However, continuous media have quite different communication needs: The continuous-media file server (see the "File servers" sidebar) must transmit and deliver data as a steady stream, especially for presentational applications, because irregularities in the dataflow will cause degradation of the audio or video quality. However, conversational applications, though highly delay sensitive, can accept a certain level of loss and data corruption in most cases.

Consequently, the usual QOS parameters for multimedia transport protocols are transport-service-data-unit maximum size, throughput, and end-to-end transit delay. Guaranteeing given values of these QOS parameters requires some kind of resource reservation, though different projects use different approaches.

The Dash approach. Anderson, Herrtwich, and Schaefer[14] based a resource reservation protocol for guaranteed performance communication in IP-based distributed systems, called Session Reservation Protocol (SRP), on the Dash resource model. This protocol allows the reservation of resources, such as CPU and network bandwidth, to achieve given delay and throughput.

The Tenet approach. Tenet provides a set of schemes and protocols for multimedia communication. It supports QOS bounds on delay, jitter, and the probability of delay violation and buffer overflow. The protocol suite includes the Real-Time Channel Administration Protocol (RCAP), which sets up the channel and reserves the required resources, and the Real-Time Internet Protocol (RIP), which schedules the packages according to the reserved resources. There are two transport protocols, the Real-Time Message Transport Protocol (RMTP), which supports message-based real-time transport between the endpoints, and the Continuous Media Transport Protocol (CMTP), which offers a stream-based interface for isochronous applications.[15] An extension to the Tenet scheme introduces Graceful Adaptation Schemes (GDS),[16] which allow either the client or the network to adopt new QOS parameters during the lifetime of an established connection.

The ST-II approach. The Experimental Internet Stream Protocol, Version 2 (ST-II)[17] is a network-layer protocol providing point-to-multipoint services. It provides facilities to negotiate and reserve resources for packet size and data rate. ST-II actually consists of two protocols: a data-forwarding protocol called ST and the ST Control Message Protocol (SCMP). SCMP controls the broadcast tree by adding and removing target addresses, and by negotiating and setting QOS parameters. ST delivers data packages only through this broadcast tree.

The HeiTS approach. The Heidelberg Transport System (HeiTS) puts the Heidelberg Transport Protocol (HeiTP) on top of an ST-II implementation. Features added to ST-II include graceful service degradation and queuing feedback for automatic synchronization between the sender and receiver to optimize the throughput and to avoid buffer overflows at the receiver side.[18] HeiTP contains four reliability classes: ignoring, discarding, indicating, and correcting corrupt data. This approach is motivated by the use of compression schemes such as MPEG for isochronous data, where corrupted data packages can have more severe consequences than in transmitting uncompressed video.

The Berkom approach. The Berkom approach provides a transport system similar to HeiTS.[19] The transport service (called the multimedia transport service, or MMTS) supports the following QOS parameters: transport service data unit maximum size, throughput, end-to-end transit delay, and the same four reliability classes as HeiTS. The multimedia transport protocol (MMTP) is also implemented on top of ST-II.

Application-layer protocols

Many application-level protocols, such as RSVP (see the "Protocol readings" sidebar), assume scalable media. For example, the approach by Delgrossi et al.[11] is also based on splitting multimedia data into separate streams. Each stream would have different QOS features, using intra- and interframe coding (see "The impact of encoding," above) to optimize the limited available bandwidth.

Other approaches provide primitives for negotiation between various components of a distributed multimedia application (see the section "QOS negotiation and renegotiation," below). The application-layer protocol for Movie Control, Access, and Management (MCAM), based on an extended X-protocol, includes among the QOS parameters reliability, speed, mode, quality, section, and direction.[20] MCAM provides primitives for setting, but not for negotiating, these parameters.

Protocol Readings

Lower level

DePrycker provides an introduction to ATM technology.

M. DePrycker, *Asynchronous Transfer Mode: Solution for Broadband ISDN*, Ellis Horwood, Chichester, England, 1993.

Damaskos and Gavras showed how the QOS parameters map from the transport layer to ATM.

S. Damaskos and A. Gavras, "Connection-Oriented Protocols over ATM: A Case Study," *Proc. IS&T/SPIE Symposium on Electronic Imaging: Science & Technology, Workshop on High-Speed Networking and Multimedia Computing*, SPIE, Bellingham, Wash., 1994, pp. 266-278.

Transport layer

For background on how the Dash system supports continuous media, consult the following.

D.P. Anderson et al., "Support for Continuous Media in the DASH System," *Proc. 10th Int'l Conf. on Distributed Systems*, CS Press, Los Alamitos, Calif., 1990, pp. 54-61.

Application layer

The Resource Reservation Protocol (RSVP), based on layered coding principles, assumes a server multicasting video over different streams, allowing clients to select streams according to their individual QOS requirements.

L. Zhang et al., "RSVP: A New Resource Reservation Protocol," *IEEE Network*, Vol. 7, No. 5, Sept. 1993, pp. 8-18.

Another resource reservation protocol that provides for dynamic changes of QOS parameters of an established channel is the Capacity-Based Session Reservation Protocol (CBSRP).

S.T.C. Chou and H. Tokuda, "System Support for Dynamic QOS Control of Continuous Media Communication," in *Network and Operating System Support for Digital Audio and Video*, P. Venkat Rangan ed., Springer-Verlag, Berlin, 1993, pp. 361-367.

Databases

Database systems are an important component of distributed multimedia systems. They provide persistent and coherent storage of multimedia objects as well as concurrent access to these objects and their components. These services should be provided in a fully distributed environment transparent to users and applications.

Information stored in a database falls into two categories: *multimedia* information, such as the multimedia objects stored and accessed by the applications, and *control* information, such as synchronization scenarios, layouts, QOS parameters, and localization rules. The system uses control information to access, deliver, and present the multimedia objects.

The database system must provide languages to define and manipulate these two different types of information. The data definition language should allow the database designer to specify the three main components of a multimedia object: its structure, content, and presentation. It must also support a powerful data model that provides

Multimedia Document Modeling

A lot of efforts are presently dedicated to multimedia document modeling, especially in the area of data models and standardization.

Data models supported by object-oriented database systems provide the essential concepts for multimedia document modeling.

T. Atwood et al., *The Object Database Standard: ODMG-93*, Morgan Kaufmann, Palo Alto, Calif., 1994.

J. Melton and A.R. Simon, *Understanding the New SQL: A Complete Guide*, Morgan Kaufmann, Palo Alto, Calif., 1993.

E. Bertino and L. Martino, "Object-Oriented Database Management Systems: Concepts and Issues," *Computer*, Vol. 24, No. 4, Apr. 1991, pp. 33-47.

R.G.G. Cattell, *Object Data Management: Object-Oriented and Extended Relational Database Systems*, Addison-Wesley, Reading, Mass., 1991.

The main ongoing efforts in the area of multimedia and hypermedia document standardization are MHEG, Hytime, and extensions of ODA.

R. Price, "MHEG: An Introduction to the Future International Standard for Hypermedia Object Interchange," *Proc. 1st ACM Int'l Conf. on Multimedia*, ACM Press, New York, pp. 121-128.

S.R. Newcomb, N.A. Kipp, and V.T. Newcomb, "The HyTime Hypermedia/Time-based Document Structuring Language," *Comm. ACM*, Vol. 34, No. 11, Nov. 1991, pp. 67-83.

R. Hunter, P. Kaijser, and F. Nielsen, "ODA: A Document Architecture for Open Systems," *Computer Comm.*, Vol. 12, No. 2, Apr. 1989, pp. 69-79.

concepts for basic objects, composite objects, and relationships. Various object models supported by existing object-oriented database management systems provide these concepts (see the "Multimedia document modeling" sidebar). Nevertheless, some of these data models must be enhanced to handle the specific characteristics of multimedia objects.

The data manipulation language should let the user insert, retrieve, modify, and delete objects in the database. Considerable work has been dedicated to query languages for multimedia databases (see the "Multimedia queries" sidebar). Most current work on database languages focuses on the definition and manipulation of multimedia information, and only to a lesser degree on defining and processing control information such as QOS parameters. However, we feel that the database manipulation language should specify the database requirements for manipulating control information, in particular the QOS parameters and localization rules, for the QOS negotiation protocol.

QOS negotiation and renegotiation

All components of a distributed system have their own QOS parameters. Some of these parameters are mutually dependent, with this dependence expressed by mappings between the system's architectural layers. An application must take all these parameters into account and negotiate values that satisfy the constraints of all components involved. Besides the initial negotiation, a distributed multimedia system must plan for

Multimedia Queries

A number of projects concern query languages for multimedia databases in particular domains, such as medical applications:

A.F. Cardenas et al., "The Knowledge-Based Object-Oriented PICQUERY+ Language," *IEEE Trans. on Knowledge and Data Eng.*, Vol. 5, No. 4, Aug. 1993, pp. 644-657.

W.W. Chu et al., "A Temporal Evolutionary Object-Oriented Data Model and its Query Language for Medical Image Management," *Proc. VLDB 92*, Morgan Kaufmann, Palo Alto, Calif., 1992, pp. 53-64.

T. Joseph and A. Cardenas, "PICQUERY : A High Level Query Language for Pictorial Database Management," *IEEE Trans. on Software Eng.*, Vol. 14, No. 5, May 1988, pp. 630-638.

and office information systems:

H. Ishikawa et al., "The Model, Language, and Implementation of an Object-Oriented Multimedia Knowledge Base Management System," *ACM Trans. on Database Systems*, Vol. 18, No. 1, Mar. 1993, pp. 1-50.

S. Christodoulakis et al., "Multimedia Document Presentation, Information Extraction, and Document Formation in MINOS: A Model and a System," *ACM Trans. on Office Information Systems*, Vol. 4, No. 4, Oct. 1986, pp. 345-383.

Each of these systems offers multimedia document retrieval optimized for images, text, or graphics.

Other articles propose ways to query multimedia databases and video databases.

F. Golshani and N. Dimitrova, "Design and Specification of EVA: A Language for Multimedia Database Systems," *Proc. 3rd Int'l Conf. on Database and Expert Systems*, Springer Verlag, Berlin, 1992, pp. 356-362.

E. Oomoto and K. Tanaka, "OVID: Design and Implementation of a Video Database System," *IEEE Trans. on Knowledge and Data Eng.*, Vol. 5, No. 4, Aug. 1993, pp. 629-643.

QOS monitoring and renegotiation as well.

We know of only a few approaches to QOS negotiation. Nahrstedt and Smith[21] presented the QOS Broker, which negotiates between application, operating system, and transport protocols. Elliot's[22] example of QOS negotiation for multimedia conferencing considered only the coding format and bandwidth of the communication network, while Kerhervé et al.[23] provided a more general approach to QOS negotiation and renegotiation that employs a three-party QOS negotiation protocol illustrated by Figure 3.

Another direction of research develops integrated approaches, including the mapping of QOS parameters between different layers. The work on a QOS architecture within the QOS-A project at the University of Lancaster covers QOS concerns from a distributed application platform to an ATM network, including parameter mappings.[24]

Conclusions

This survey differs from others in examining QOS parameters in all components of distributed multimedia applications, in particular communication protocols, coding schemes, operating systems, continuous file servers, and databases. We see future work in the integration of QOS within distributed application platforms or middleware such as OMG's Common Object Request Broker Architecture (CORBA) or OSF's Distributed Computing Environment (DCE).[25] Work in this area is on real-time extensions to ANSAware[26,27] or within the Touring machine project.[28] OSF released recently a real-time Mach-based operating system. We also envision including QOS negotiation protocols in the World Wide Web's http protocol. **MM**

Acknowledgments

We are particularly grateful to our colleagues of the QOS subproject, Rachida Dssouli, Abdelhakim Hafid, Alain Bibal, Thibault Burdin de Saint-Martin, Quoc Vu, and Luc Ouedraogo, for their contribution to the work presented here. We would like also to thank our partners in the CITR project, Eric Dubois, Niklas Georganas, Gerald Neufeld, Tamar Özsu, and Johny Wong, for fruitful discussions on this project. Special thanks go to David Jackson. This work was supported by a grant from the Canadian Institute for Telecommunication Research (CITR), under the Networks of Centers of Excellence Program of the Canadian Government.

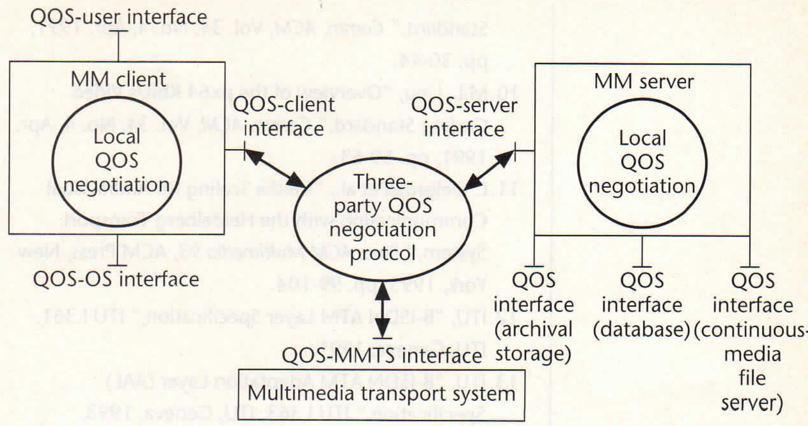


Figure 3. The Kerhervé approach applies three-party QOS negotiation to presentation applications.

References

1. "Reference Model of Open Distributed Processing—Part 3: Prescriptive Model," Int'l Standard 10746-3, ITU-T Recommendation X.903, ITU-ISO, Geneva, 1995.
2. G. Kalkbrenner et al., "Quality of Service (QOS) in Distributed Hypermedia Systems," *Proc. 2nd Int'l Workshop on Principles of Document Processing*, 1994.
3. R.G. Herrtwich, "The Role of Performance, Scheduling, and Resource Reservation in Multimedia Systems," in *Operating Systems of the 90s and Beyond*, A. Karshmer and J. Nehmer, eds. Springer-Verlag, Berlin, 1991, pp. 279-284.
4. T. Nakajima, T. Kitayama, and H. Tokuda, "Experiments with Real-Time Servers in Real-Time Mach," tech. report, School of Computer Science, Carnegie Mellon Univ., Pittsburgh, 1993.
5. R.B. Dannenberg et al., "Performance Measurements of the Multimedia Testbed on Mach 3.0: Experience Writing Real-Time Device Drivers, Servers, and Applications," Tech. Report #CMU-CS-93-205, School of Computer Science, Carnegie Mellon Univ., Pittsburgh, 1993.
6. C.M. Mercer, S. Savage, and H. Tokuda, "Procesor Capacity Reserves for Multimedia Operating Systems," Tech. Report #CMU-CS-93-157, School of Computer Science, Carnegie Mellon Univ., Pittsburgh, 1993.
7. K. Nahrstedt and J. Smith, "Experimental Study of Issues in End-to-End QOS," tech. report, Computer Science Dept., Univ. of Pennsylvania, 1994.
8. J.P. Parsons, Jr., "Digital Video Architecture for OS/2 2.1," *Proc. IS&T/SPIE Symp. on Electronic Imaging: Science & Technology, Workshop on High-Speed Networking and Multimedia Computing*, SPIE, Bellingham, Wash., 1994, pp. 19-30.
9. G.K. Wallace, "The JPEG Still Picture Compression

Standard," *Comm. ACM*, Vol. 34, No. 4, Apr. 1991, pp. 30-44.

10 M.L. Liou, "Overview of the px64 Kbit/s Video Coding Standard," *Comm. ACM*, Vol. 34, No. 4, Apr. 1991, pp. 59-63.

11. L. Delgrossi et al., "Media Scaling for Audiovisual Communication with the Heidelberg Transport System," *Proc. ACM Multimedia 93*, ACM Press, New York, 1993, pp. 99-104.

12. ITU, "B-ISDN ATM Layer Specification," ITU I.361, ITU, Geneva, 1991.

13. ITU, "B-ISDN ATM Adaptation Layer (AAL) Specification," ITU I.363, ITU, Geneva, 1993.

14. D. Anderson, R. Herrtwich, and C. Schaefer, "SRP: A Resource Reservation Protocol for Guaranteed-Performance Communications in the Internet," tech. report, The Int'l Computer Science Inst., Berkeley, Calif., 1991.

15. D. Ferrari, A. Banerjea, and H. Zhang, "Network Support For Multimedia," Tech. Report TR-92-072, The Int'l Computer Science Inst., Berkeley, Calif., 1992.

16. C.J. Parris, G. Ventre, and H. Zhang, "Graceful Adaption of Guaranteed Performance Service Connections," Tech. Report TR-93-011, The Int'l Computer Science Inst., Berkeley, Calif., 1993.

17. C. Topolcic, "Experimental Internet Stream Protocol, Version 2 (ST-II)," Internet RFC 1190, lPublisher? Publisher's location?l, 1990.

18. L. Delgrossi, R.G. Herrtwich, and F.O. Hoffmann, "An Implementation of ST-II for the Heidelberg Transport System," *Proc. IEEE Globcom 92*, CS Press, Los Alamitos, Calif., 1992.

19. S. Boecking et al., "The Berkom Multimedia Transport System," *Proc. IS&T/SPIE Symp. on Electronic Imaging: Science & Technology, Workshop on High-Speed Networking and Multimedia Computing*, SPIE, Bellingham, Wash., 1994, pp. 256-265.

20. R. Keller and W. Effelsberg, "MCAM: An Application Layer Protocol for Movie Control, Access, and Management," *Proc. 1st ACM Int'l Conf. on Multimedia*, ACM Press, New York, 1993, pp. 21-30.

21. K. Nahrstedt and J.M. Smith, "Application-Driven Approach to Networked Multimedia System," *Proc. 18th Conference on Local Computer Networks*, CS Press, Los Alamitos, Calif., 1993, pp. 361-368.

22. C. Elliot, "High-Quality Multimedia Conferencing Through a Long-Haul Packet Network," *Proc. 1st ACM Int'l Conf. on Multimedia*, ACM Press, New York, 1993, pp. 91-98.

23. B. Kerhervé et al., "On Distributed Multimedia Presentational Applications: Functional and Computational Architecture and QOS Negotiation," *Proc. 4th Int'l Workshop on Protocols for High-Speed Networks*, Chapman & Hall, London, 1994, pp. 1-17.

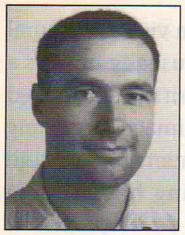
24. A. Campbell et al., "Integrated Quality of Service for Multimedia Communications," *Proc. IEEE InfoCom*, CS Press, Los Alamitos, Calif., 1993, pp. 732-739.

25. A. Schill et al., "A Quality of Service Abstraction Tool for Advanced Distributed Applications" *Open Distributed Processing II*, Chapman & Hall, London, 1995, in press.

26. C. Nicolaou, *A Distributed Architecture for Multimedia Communication Systems*, doctoral dissertation, Univ. of Cambridge, UK, 1991.

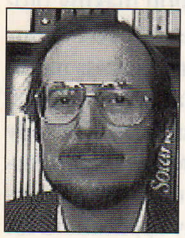
27. G. Coulson et al., "Supporting the Real-Time Requirements of Continuous Media in Open Distributed Processing," Tech. Report MPG-92-35, Univ. of Lancaster, 1992.

28. M. Arango et al., "Touring Machine: A Software Infrastructure to Support Multimedia Communications," *ACM Computer Comm. Rev.*, Vol. 22, Mar. 1992, pp. 53-54.

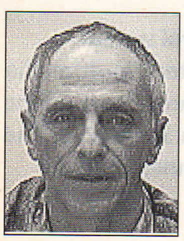


Andreas Vogel works as senior research scientist at the Cooperative Research Centre for Distributed Systems Technology (DSTC), Brisbane, Australia. He received his MSc from Humboldt-Universität zu Berlin, Berlin, Germany in 1990. Current work includes type management, service trading, and quality of service negotiation in distributed systems. He is actively involved in the standardization of the Reference Model for Open Distributed Processing.

Brigitte Kerhervé is a professor with the Computer Science Department in Université du Québec á Montréal, Canada. She received a PhD in computer science from the University of Paris VI, France in 1986. Her research interests include metadata for multimedia databases, as well as object oriented and distributed database systems to support distributed multimedia applications. She is a member of the ACM and the IEEE Computer Society.



Gregor v. Bochmann holds the Hewlett-Packard-NSERC-CITI chair of industrial research on communication protocols in the Département d'informatique et de recherche opérationnelle at the Université de Montréal. He received the Diploma degree in physics from the University of Munich in 1968 and a PhD from McGill University, Montréal, in 1971. He has been actively involved in the standardization of formal description techniques for OSI. His present work is aimed at design methods for communication protocols and distributed systems.



Jan Gecsei is professor of computer science at the Université de Montréal. He earned his MSEE at the Czech Techical University and his PhD in computer science at the Acadamy of Sciences in Prague. His research interests include distributed multimedia systems and intelligent tutoring systems. He is a Senior Member of IEEE.

Readers can contact Vogel at DSTC, Level 7 Gehrmann Laboratories, University of Queensland, Q 4072, Australia, e-mail andreas@dstc.edu.au, home page http://www.dstc.edu.au/AU/staff/ andreasvogel.html.